

Amendments to the Claims

Kindly amend claims 1, 3, 10 & 12, and cancel claims 4, 9, 13 & 18-20 (without prejudice), as set forth below. All pending claims are reproduced below, with changes in the amended claims shown by underlining (for added matter) and strikethrough/double brackets (for deleted matter).

1. (Currently Amended) A method of transferring executable program code between computer processes, the method comprising:

providing by a sender an object which comprises a hashtable, the hashtable having at least one set of elements, one element of the at least one set of elements comprising executable program code, the providing comprising:

creating an empty hashtable;

integrating executable program code into the hashtable; and

serializing the hashtable into a serialized data object for transport to the receiver computer process;

wherein the executable program code comprises logic which employs as data input ~~at least one other element of the hashtable;~~ and

transferring the object from ~~[[a]]~~ the sender computer process to a receiver computer process, retrieving the executable program code from the hashtable and invoking the executable program code with the ~~at least one other element of the hashtable as~~ the only data input thereto, wherein data is added to the hashtable by the receiver prior to invoking of the executable program code retrieved from the hashtable with the hashtable as the only data input thereto.

2. (Original) The method of claim 1, wherein the at least one set of elements comprises multiple tuples, each tuple comprising a first element and a second element, and wherein the second element of at least one tuple comprises the program code, and the second element of at least one other tuple comprises data relevant to the program code.

3. (Currently Amended) The method of claim 1, wherein the providing comprises providing the object as a serialized data object and transporting the serialized data object from the sender computer process to the receiver computer process, and wherein the method further comprises deserializing the serialized data object at the receiver computer process to obtain the hashtable, scanning the hashtable for executable program code, and invoking the executable program code with [[the]] at least other element of the hashtable as data input thereto.

4. (Canceled).

5. (Previously Presented) The method of claim 1, wherein the providing comprises providing multiple serialized objects, each serialized object having a different hashtable therein, and transporting a first serialized object from a first sender computer process to the receiver computer process and transporting a second serialized object from a second sender computer process to the receiver computer process, and deserializing the first serialized object and the second serialized object at the receiver computer process to obtain a first hashtable and a second hashtable.

6. (Original) The method of claim 5, wherein the first sender computer process, the second sender computer process, and the receiver computer process are on different computing units.

7. (Previously Presented) The method of claim 5, further comprising merging the first hashtable received from the first sender computer process and the second hashtable received from the second sender computer process at the receiver computer process into a common hashtable, and iterating through the common hashtable for executable program code to be invoked using the common hashtable as the only data input thereto.

8. (Previously Presented) The method of claim 7, further comprising adding data to the common hashtable at the receiver computer process, the data being relevant to executable program code in the common hashtable and being added prior to invoking the executable program code using as data input only the common hashtable.

9. (Canceled).

10. (Currently Amended) A system for transferring executable program code between computer processes, said system comprising:

means for providing by a sender an object which comprises a hashtable, the hashtable having at least one set of elements, one element of the at least one set of elements comprising executable program code, the means for providing comprising means for:

creating an empty hashtable;

integrating executable program code into the hashtable; and

serializing the hashtable into a serialized data object for transport to the receiver computer process;

wherein the executable program code comprises logic which employs as data input ~~at least one other element of the hashtable;~~ and

means for transferring the object from ~~[[a]]~~ the sender computer process to a receiver computer process, for retrieving the executable program code from the hashtable and for invoking the executable program code with the ~~at least one other element of the hashtable as the only~~ data input thereto, wherein data is added to the hashtable by the receiver prior to invoking of the executable program code retrieved from the hashtable with the hashtable as the only data input thereto.

11. (Original) The system of claim 10, wherein the at least one set of elements comprises multiple tuples, each tuple comprising a first element and a second element, and wherein the second element of at least one tuple comprises the program code, and the second element of at least one other tuple comprises data relevant to the program code.

12. (Currently Amended) The system of claim 10, wherein the means for providing comprises means for providing the object as a serialized data object and for transporting the serialized data object from the sender computer process to the receiver computer process, and wherein the system further comprises means for deserializing the serialized data object at the receiver computer process to obtain the hashtable, means for scanning the hashtable for executable program code, and means for invoking the executable program code with ~~[[the]]~~ at least one other element of the hashtable as data input thereto.

13. (Canceled).

14. (Previously Presented) The system of claim 10, wherein the means for providing comprises means for providing multiple serialized objects, each serialized object having a different hashtable therein, and wherein the system further comprises means for transporting a first serialized object from a first sender computer process to the receiver computer process and for transporting a second serialized object from a second sender computer process to the receiver computer process, and means for deserializing the first serialized object and the second serialized object at the receiver computer process to obtain a first hashtable and a second hashtable.

15. (Original) The system of claim 14, wherein the first sender computer process, the second sender computer process, and the receiver computer process are on different computing units.

16. (Previously Presented) The system of claim 14, further comprising means for merging the first hashtable received from the first sender computer process and the second hashtable received from the second sender computer process at the receiver computer process into a common hashtable, and means for iterating through the common hashtable for executable program code to be invoked using the common hashtable as the only data input thereto.

17. (Previously Presented) The system of claim 16, further comprising means for adding data to the common hashtable at the receiver computer process, the data being relevant to executable program code in the common hashtable and being added prior to invoking the executable program code using as data input only the common hashtable.

18-20. (Canceled).

* * * * *